Amendments to the Claims:

Please amend Claims 1, 14, 27, 31-35, 37, and 38 as follows:

1. (currently amended) A computer-readable medium having stored therein a data structure in the form of a language module for storing data strings used by a display management module to display information on a display terminal, wherein said data structure comprises a string data area stored in a range of memory addresses in the computer-readable medium, wherein said string data area includes a plurality of data strings to be displayed by the display management module, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters, wherein said characters are stored in said string data area based on an escape code having a value selected such that all of the standard ASCII, extended ASCII, and double characters can be properly stored in the string data area, and wherein characters in a data string that are standard ASCII and extended ASCII characters having ASCII codes less than a selected escape code are stored by their ASCII representations in said string data area, while extended ASCII characters and standard ASCII characters, if any, having ASCII codes at least as great as the selected escape code and ASCII characters that identify double byte characters are encoded and the encoded values are stored in said string data area.

2. (original) A computer-readable medium according to Claim 1, wherein said data structure further comprises a font module containing font data for displaying 256 standard and extended ASCII characters.

3. (original) A computer-readable medium according to Claim 2, wherein said data structure further comprises an extended ASCII font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to extended ASCII characters that are not displayable with the extended ASCII character data stored in said font module.

4. (original) A computer-readable medium according to Claim 2, wherein said data structure further comprises a double byte character font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to double byte characters.

5. (original) A computer-readable medium according to Claim 4, wherein said double byte character font data area includes only font data for the double byte characters present in data strings to be displayed by the display management module to thereby minimize data storage space.

6. (original) A computer-readable medium according to Claim 4, wherein double byte characters are sequentially encoded with 16 bit values and the code values representing the double byte characters are stored in said string data area, and wherein font data associated with the double byte characters is stored in the double byte character font data area.

7. (original) A computer-readable medium according to Claim 2, wherein extended ASCII characters and standard ASCII characters, if any, having ASCII codes at least as great as the selected escape code are encoded in the string data area as 16 bit codes, wherein each 16 bit code comprises an escape code proceeded by the ASCII code of the standard ASCII and extended ASCII character.

8. (original) A computer-readable medium according to Claim 6, wherein double byte characters are sequentially encoded into 16 bit codes, wherein the first double byte character is represented by a 16 bit code having a first byte that is one value greater than the escape code and a second byte equal to zero and the remaining unique double byte characters are encoded with sequential 16 bit code values.

9. (original) A computer-readable medium according to Claim 8, wherein the selected escape code is E0 hexadecimal, wherein double byte characters are sequentially encoded and stored in said string data area such that the first double byte character is represented by the 16 bit code value E100 in hexadecimal and the remaining unique double byte characters are encoded

with sequential 16 bit values from E101 to FFFF hexadecimal, and wherein the extended ASCII characters having ASCII codes at least as great as the selected escape code are encoded in said string data area as 16 bit codes with the selected escape code as the first byte and the ASCII code for the extended ASCII character as the second byte.

10. (original) A computer-readable medium according to Claim 1, wherein said string data area includes data strings stored in a particular language, and wherein said data structure further comprises a header associated with said data string area that is stored in a range of memory addresses, wherein said header indicates the language with which the data strings are stored in said string data area.

11. (original) A computer-readable medium according to Claim 2, wherein said data structure further comprises a header having data stored therein indicating whether said data structure further includes an extended ASCII font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to extended ASCII characters that are not displayable with the extended ASCII character font data stored in said font module.

12. (original) A computer-readable medium according to Claim 11, wherein said header further indicates whether said data structure comprises a double byte character font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to characters that are double byte characters.

13. (original) A computer-readable medium according to Claim 1, wherein said data structure further comprises a string pointer table stored in a range of memory addresses including pointers indicating the location of data strings in said string data area.

14. (currently amended) A method for creating a data structure in a computer-readable medium in the form of a language module for storing data strings used by a display management module to display information on a display terminal, wherein said method comprises the steps of:

creating a string data area in a range of memory addresses in the computer-readable medium for storing each of the data strings;

determining the number of standard ASCII, extended ASCII, and double byte characters to be stored in the string data area;

selecting a value of an escape code such that all of the standard ASCII, extended ASCII, and double characters can be properly stored in the string data area;

analyzing each character of each data string, wherein each character is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

encoding extended ASCII characters and standard ASCII characters, if any, having ASCII values at least as great as a selected escape code and ASCII characters that identify double byte characters with encoded values;

storing each data string in the string data area, wherein for each data string said storing step stores standard ASCII characters and extended ASCII characters having ASCII codes less than the selected escape code by their ASCII representations and stores extended ASCII characters and standard ASCII characters, if any, having ASCII values greater than or equal to the selected escape code and ASCII characters that identify double byte characters by their encoded values as determined in said encoding step.

15. (original) A method according to Claim 14, wherein said creating step further creates a font module in the data structure containing font data for displaying 256 standard and extended ASCII characters.

16. (original) A method according to Claim 15, wherein said creating step further comprises creating an extended ASCII font data area in a range of memory addresses in the computer-readable medium, wherein the extended ASCII font data area includes font data related to extended ASCII characters that are not displayable with the extended ASCII character font data stored in the font module.

17. (original) A method according to Claim 15, wherein said creating step further comprises creating a double byte character font data area in a range of memory addresses in the computer-readable medium, wherein the double byte character font data area includes font data related to characters that are double byte characters.

18. (original) A method according to Claim 17, wherein said storing step only stores font data in the double byte character font data area for the double byte characters present in the data strings to be displayed by the display management module instead of all possible double byte characters to thereby minimize data storage space.

19. (original) A method according to Claim 17, wherein said encoding step sequentially encodes double byte characters with 16 bit codes, wherein said storing step stores the code values representing the double byte characters in the string data area, and wherein said storing step further stores font data associated with the double byte characters in the double byte character font data area.

20. (original) A method according to Claim 15, wherein said encoding step encodes extended ASCII characters and standard ASCII characters, if any, having ASCII codes at least as great as the selected escape code with 16 bit codes, wherein each 16 bit code comprises an escape code proceeded by the ASCII code of the extended ASCII or standard ASCII character..

21. (original) A method according to Claim 19, wherein said encoding step sequentially encodes double byte characters into 16 bit codes, wherein the first double byte character is represented by a 16 bit code having a first byte that is one value greater than the escape code and a second byte equal to zero and the remaining unique double byte characters are encoded with sequential 16 bit code values.

22. (original) A method according to Claim 21, wherein the escape code is E0 hexadecimal, wherein said encoding step sequentially encodes double byte characters such that the first double byte character is represented by the 16 bit encoded value E100 hexadecimal and the remaining unique double byte characters are encoded with sequential 16 bit values from

E101 to FFFF hexadecimal, and wherein said encoding step encodes extended ASCII characters having ASCII codes at least as great as the selected escape code as 16 bit codes with the selected escape code as the first byte and the ASCII code for the extended ASCII character as the second byte.

23. (original) A method according to Claim 15, wherein the string data area includes data strings stored in a particular language, and wherein said creating step further comprises creating a header associated with the data string area in a range of memory addresses for indicating the language in which the data strings are stored in the string data area.

24. (original) A method according to Claim 23, wherein said storing step stores in the header an indication as to whether the data structure further includes an extended ASCII font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to extended ASCII characters that are not displayable with the extended ASCII character font data stored in the font module.

25. (original) A method according to Claim 23, wherein said storing step stores in the header an indication as to whether the data structure further comprises a double byte character font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to characters that are double byte characters.

26. (original) A method according to Claim 14, wherein said creating step further creates a string pointer table in a range of memory addresses, wherein the string pointer table includes pointers indicating the location of data strings in the string data area.

27. (currently amended) A computer program product for creating a data structure in the form of a language module for storing data strings used by a display management module to display information on a display terminal, wherein the computer program product comprises:

a computer-readable storage medium having computer readable program code means embodied in said medium, said computer-readable program code means comprising:

first computer instruction means for creating a string data area in a range of memory addresses in the computer-readable medium for storing each of the data strings;

second computer instruction means for determining a value of an escape code that has been selected such that all of the standard ASCII, extended ASCII, and double characters can be properly stored in the string data area;

~~second~~ third computer instruction means for analyzing each character of each data string, wherein each character is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

~~third~~ fourth computer instruction means for encoding extended ASCII characters and standard ASCII characters, if any, having ASCII values at least as great as the selected escape code and ASCII characters that identify double byte characters with encoded values;

~~fourth~~ fifth computer instruction means for storing each data string in the string data area, wherein for each data string said fourth computer instruction means stores standard ASCII characters and extended ASCII characters having ASCII codes less than the selected escape code by their ASCII representations and stores extended ASCII characters and standard ASCII, if any, having ASCII values at least as great as the selected escape code and ASCII characters double byte characters by their encoded values as determined in said third computer instruction means.

28. (previously presented) A computer program product according to Claim 27, wherein said first computer instruction means creates a font module in the data structure containing font data for displaying 256 standard and extended ASCII characters.

29. (original) A computer program product according to Claim 28, wherein said first computer instruction means further comprises creating an extended ASCII font data area in a range of memory addresses in the computer-readable medium, wherein the extended ASCII font data area includes font data related to extended ASCII characters that are not displayable with the extended ASCII character font data stored in the font module.

30. (original) A computer program product according to Claim 28, wherein said first computer instruction means further comprises creating a double byte character font data area in a range of memory addresses in the computer-readable medium, wherein the double byte character font data area includes font data related to characters that are double byte characters.

31. (currently amended) A computer program product according to Claim 30, wherein said fifth fourth computer instruction means only stores font data in the double byte character font data area for the double byte characters present in the data strings to be displayed by the display management module instead of all possible double byte characters to thereby minimize data storage space.

32. (currently amended) A computer program product according to Claim 30, wherein said fourth third computer instruction means sequentially encodes double byte characters with 16 bit codes, wherein said fifth fourth computer instruction means stores the code values representing the double byte characters in the string data area, and wherein said fifth fourth computer instruction means stores font data associated with the double byte characters in the double byte character font data area.

33. (currently amended) A computer program product according to Claim 28, wherein said fourth third computer instruction means encodes extended ASCII characters and standard ASCII characters, if any, having ASCII codes at least as great as the selected escape code with 16 bit codes, wherein each 16 bit code comprises an escape code proceeded by the ASCII code of the extended ASCII or standard ASCII character.

34. (currently amended) A computer program product according to Claim 32, wherein said fourth third computer instruction means sequentially encodes double byte characters into 16 bit codes, wherein the first double byte character is represented by a 16 bit code having a first byte that is one value greater than the escape code and a second byte equal to zero and the remaining unique double byte characters are encoded with sequential 16 bit code values.

35. (currently amended) A computer program product according to Claim 34, wherein the escape code is E0 hexadecimal, wherein said fourth ~~third~~ computer instruction means sequentially encodes double byte characters such that the first double byte character is represented by the 16 bit encoded value E100 hexadecimal and the remaining unique double byte characters are encoded with sequential 16 bit values from E101 to FFFF hexadecimal, and wherein said fourth ~~third~~ computer instruction means encodes extended ASCII characters having ASCII codes at least as great as the selected escape code as a 16 bit value having the selected escape code as the first byte and the code for the extended ASCII character as the second byte.

36. (original) A computer program product according to Claim 27, wherein the string data area includes data strings stored in a particular language, and wherein said first computer instruction means further comprises creating a header associated with the data string area in a range of memory addresses for indicating the language in which the data strings are stored in the string data area.

37. (currently amended) A computer program product according to Claim 36, wherein said fifth ~~fourth~~ computer instruction means stores in the header an indication as to whether the data structure further includes an extended ASCII font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to extended ASCII characters that are not displayable with the extended ASCII character font data stored in the font module.

38. (currently amended) A computer program product according to Claim 36, wherein said fifth ~~fourth~~ computer instruction means stores in the header an indication as to whether the data structure further comprises a double byte character font data area stored in a range of memory addresses in the computer-readable medium for storing font data related to characters that are double byte characters.

39. (original) A computer program product according to Claim 27, wherein said first computer instruction means further creates a string pointer table in a range of memory addresses, wherein the string pointer table includes pointers indicating the location of data strings in the string data area.